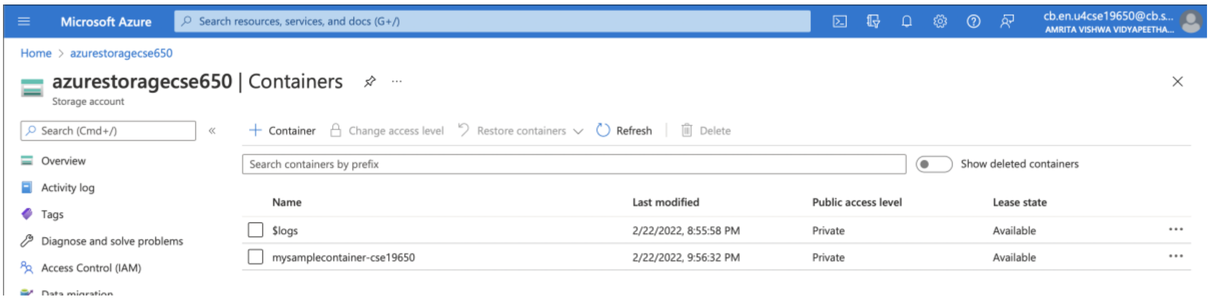
19CSE445 – CLOUD COMPUTING

22.02.2022

SUBMITTED BY,
SANDEEP RAJAKRISHNAN
CB.EN.U4CSE19650
CSE – F

Assignment 4 – Storing Files in Azure

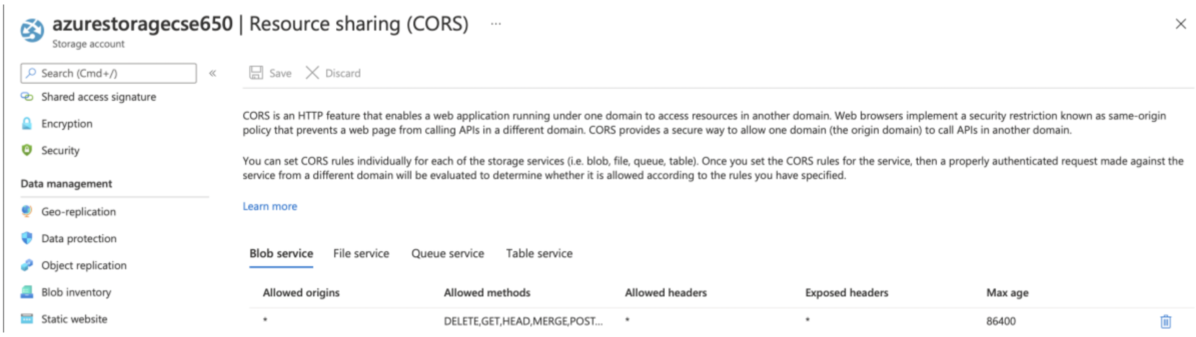
1. Create Storage Account


Microsoft Azure
Search resources, services, and docs (G+/)
cb.en.u4cse19650@cb.s...
AMRITA VISHWA VIDYAPEETHA...
Home > azurestoragecse650

azurestoragecse650 | Containers
Storage account

	Name	Last modified	Public access level	Lease state	
□	$logs	2/22/2022, 8:55:58 PM	Private	Available	···
□	mysamplecontainer-cse19650	2/22/2022, 9:56:32 PM	Private	Available	···

2. Configure CORS


azurestoragecse650 | Resource sharing (CORS)
Storage account

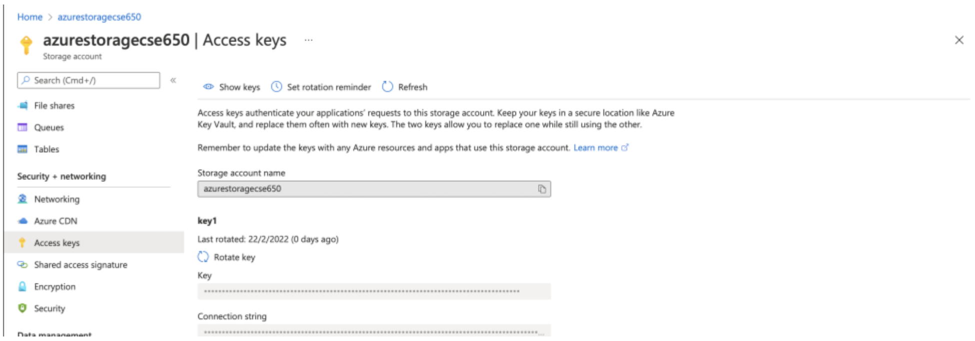
CORS is an HTTP feature that enables a web application running under one domain to access resources in another domain. Web browsers implement a security restriction known as same-origin policy that prevents a web page from calling APIs in a different domain. CORS provides a secure way to allow one domain (the origin domain) to call APIs in another domain.

You can set CORS rules individually for each of the storage services (i.e. blob, file, queue, table). Once you set the CORS rules for the service, then a properly authenticated request made against the service from a different domain will be evaluated to determine whether it is allowed according to the rules you have specified.

Learn more

Allowed origins	Allowed methods	Allowed headers	Exposed headers	Max age
*	DELETE,GET,HEAD,MERGE,POST...	*	*	86400

3. Copy AZURE_STORAGE_CONNECTION_STRING from Access Keys



azurestoragecse650 | Access keys
Storage account

Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location like Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.

Remember to update the keys with any Azure resources and apps that use this storage account. Learn more

Storage account name: azurestoragecse650

key1

Last rotated: 22/2/2022 (0 days ago)

Rotate key

Key: ••

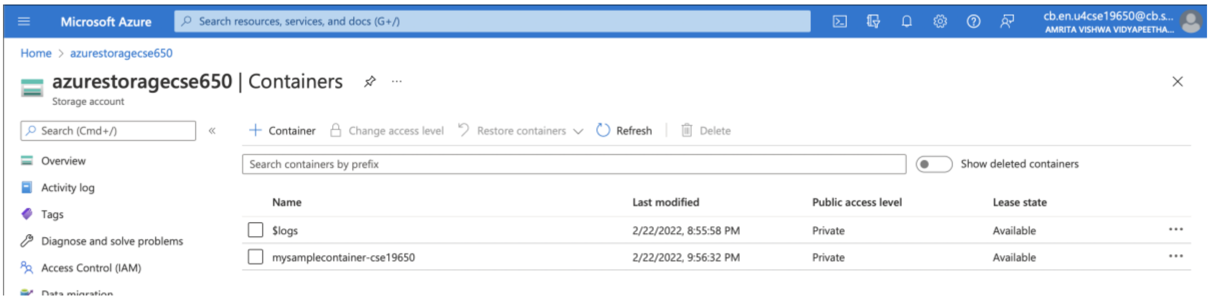
Connection string: ••...

4. Create environment variable file .env with connection string

```
AZURE_STORAGE_CONNECTION_STRING=YOUR_CONNECTION_STRING
```

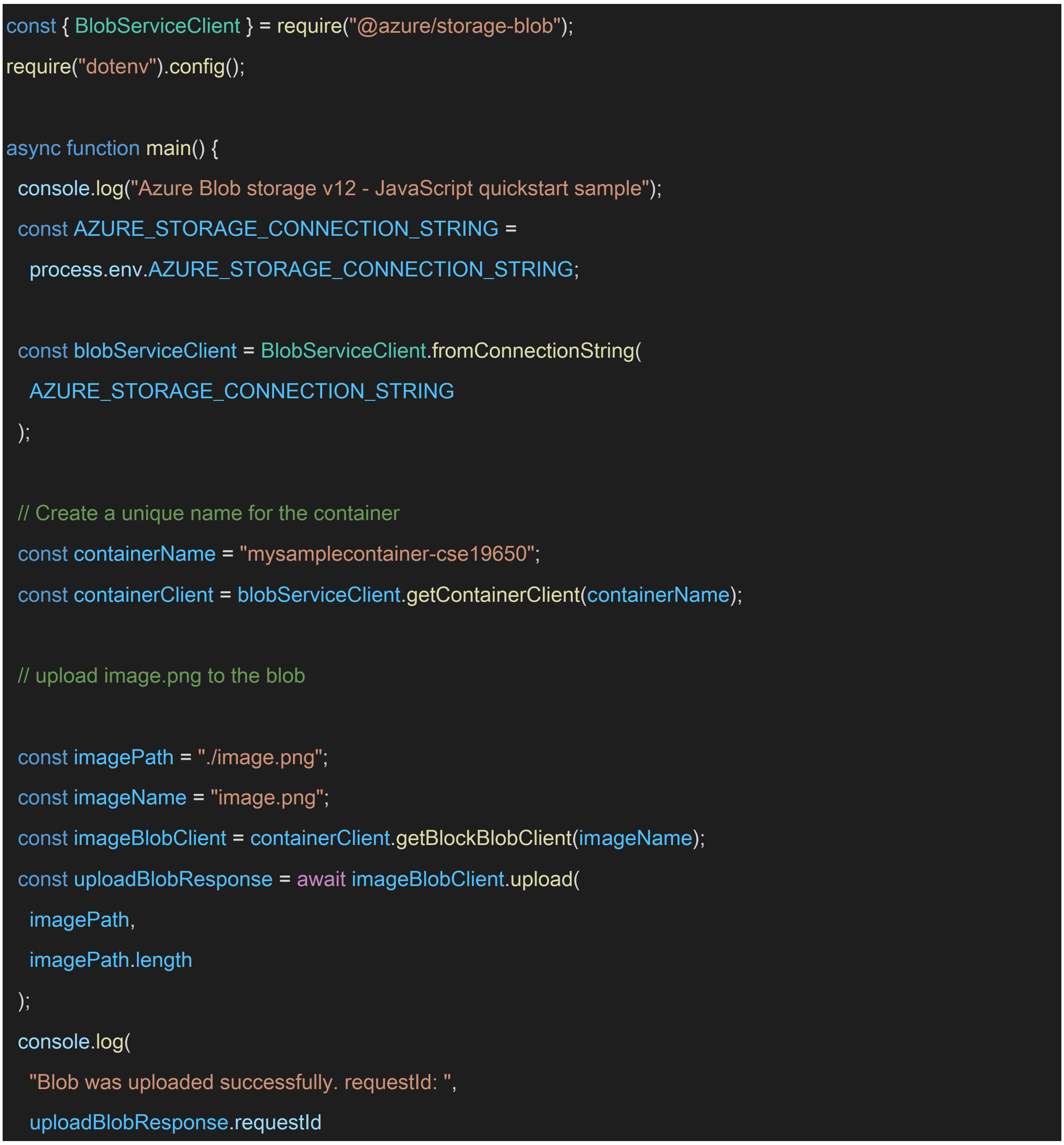
5. Creating a container with node js

```
const { BlobServiceClient } = require('@azure/storage-blob');
require('dotenv').config();


async function main() {
   console.log('Azure Blob storage v12 - JavaScript quickstart sample');
   const AZURE_STORAGE_CONNECTION_STRING =
process.env.AZURE_STORAGE_CONNECTION_STRING;

   const blobServiceClient =
BlobServiceClient.fromConnectionString(AZURE_STORAGE_CONNECTION_STRING);

// Create a unique name for the container
   const containerName = 'mysamplecontainer-cse19650';

   console.log('\t', containerName);

// Get a reference to a container
   const containerClient = blobServiceClient.getContainerClient(containerName);

// Create the container
   const createContainerResponse = await containerClient.create();
   console.log("Container was created successfully. requestId: ", createContainerResponse.requestId);

}

main().then(() => console.log('Done')).catch((ex) => console.log(ex.message));
```

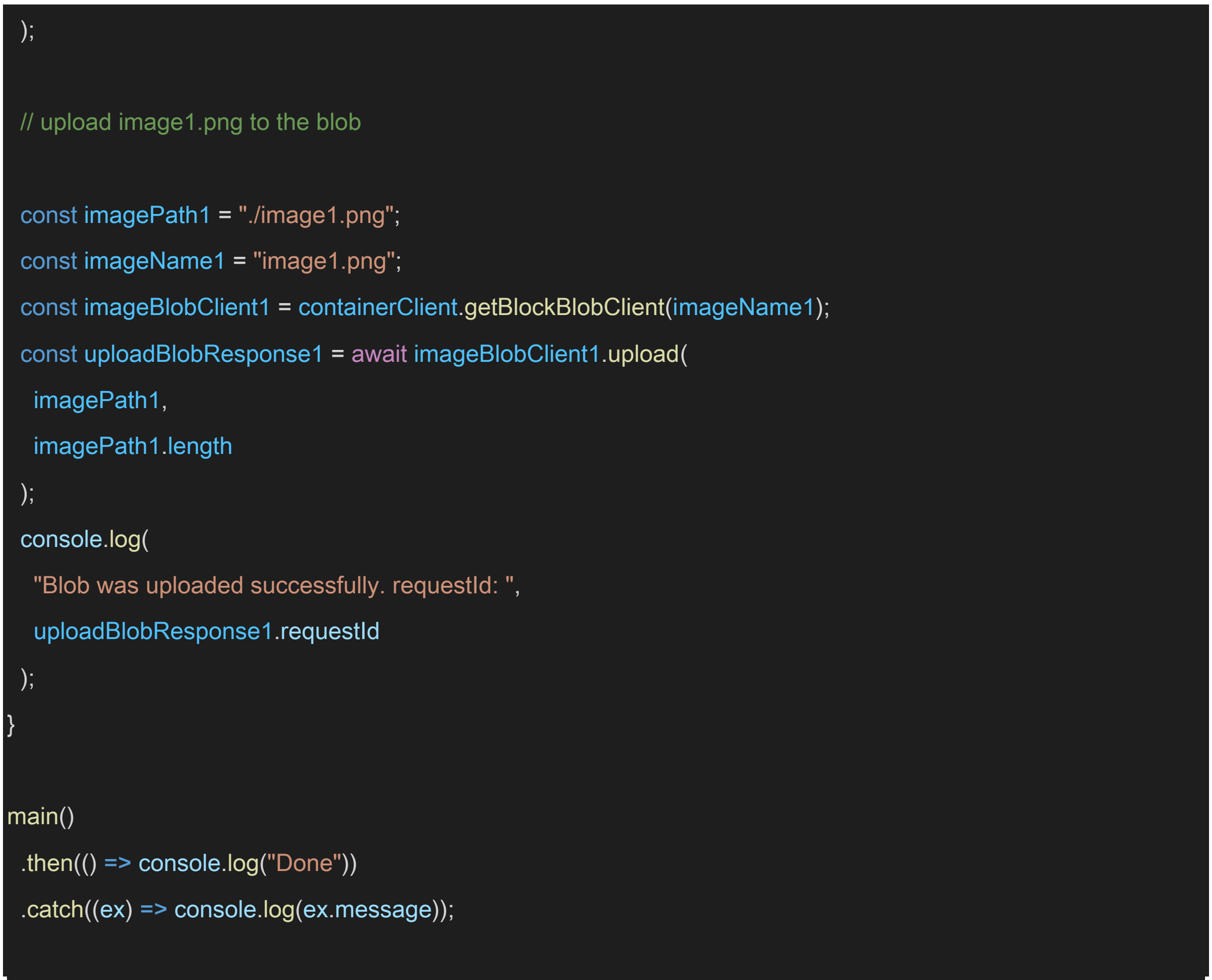


Name	Last modified	Public access level	Lease state
$logs	2/22/2022, 8:55:58 PM	Private	Available
mysamplecontainer-cse19650	2/22/2022, 9:56:32 PM	Private	Available

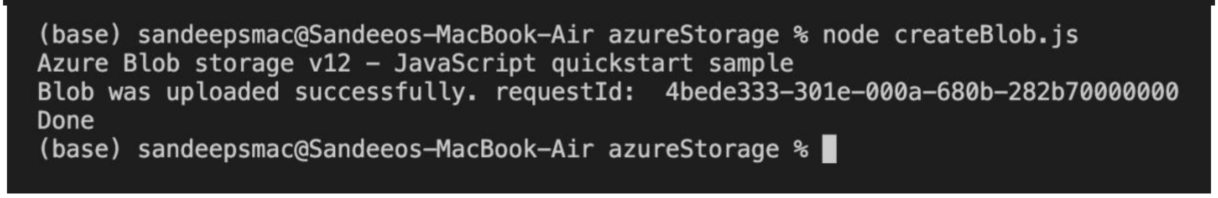
6. Creating blob and uploading two images



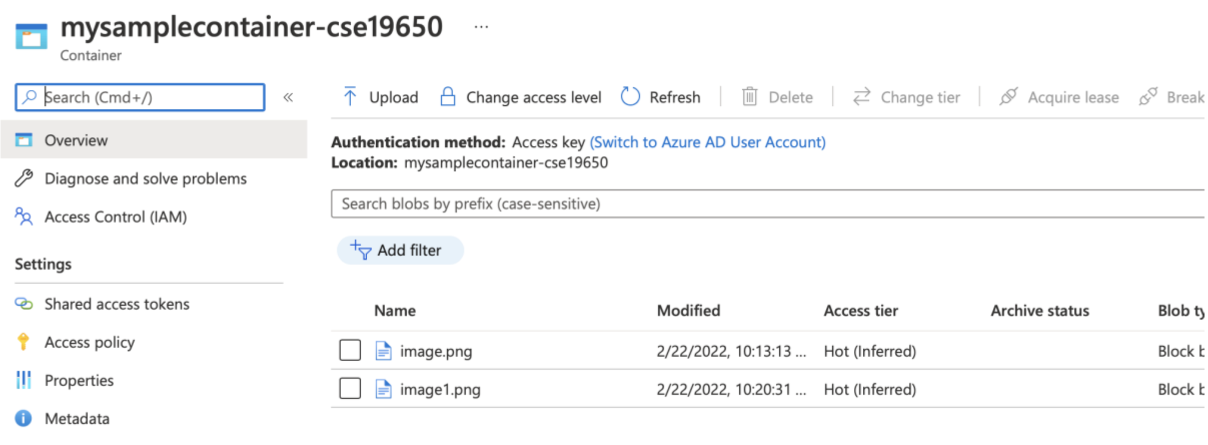
```
const { BlobServiceClient } = require("@azure/storage-blob");
require("dotenv").config();

async function main() {
  console.log("Azure Blob storage v12 - JavaScript quickstart sample");
  const AZURE_STORAGE_CONNECTION_STRING =
    process.env.AZURE_STORAGE_CONNECTION_STRING;

  const blobServiceClient = BlobServiceClient.fromConnectionString(
    AZURE_STORAGE_CONNECTION_STRING
  );

  // Create a unique name for the container
  const containerName = "mysamplecontainer-cse19650";
  const containerClient = blobServiceClient.getContainerClient(containerName);

  // upload image.png to the blob

  const imagePath = "./image.png";
  const imageName = "image.png";
  const imageBlobClient = containerClient.getBlockBlobClient(imageName);
  const uploadBlobResponse = await imageBlobClient.upload(
    imagePath,
    imagePath.length
  );
  console.log(
    "Blob was uploaded successfully. requestId: ",
    uploadBlobResponse.requestId
```



```
 );

 // upload image1.png to the blob

 const imagePath1 = "./image1.png";
 const imageName1 = "image1.png";
 const imageBlobClient1 = containerClient.getBlockBlobClient(imageName1);
 const uploadBlobResponse1 = await imageBlobClient1.upload(
   imagePath1,
   imagePath1.length
 );
 console.log(
   "Blob was uploaded successfully. requestId: ",
   uploadBlobResponse1.requestId
 );
}

main()
 .then(() => console.log("Done"))
 .catch((ex) => console.log(ex.message));
```



```
(base) sandeepsmac@Sandeeos-MacBook-Air azureStorage % node createBlob.js
Azure Blob storage v12 - JavaScript quickstart sample
Blob was uploaded successfully. requestId:  4bede333-301e-000a-680b-282b70000000
Done
(base) sandeepsmac@Sandeeos-MacBook-Air azureStorage %
```

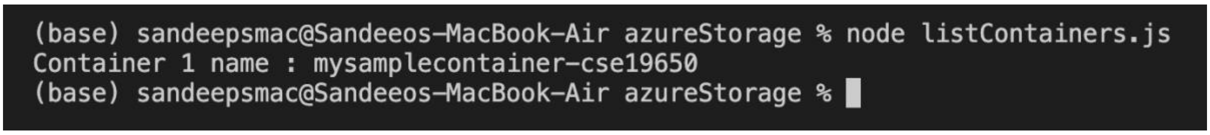


Authentication method: Access key (Switch to Azure AD User Account)
Location: mysamplecontainer-cse19650

	Name	Modified	Access tier	Archive status	Blob ty
□	image.png	2/22/2022, 10:13:13 ...	Hot (Inferred)		Block b
□	image1.png	2/22/2022, 10:20:31 ...	Hot (Inferred)		Block b

7. List all the containers in Storage Account

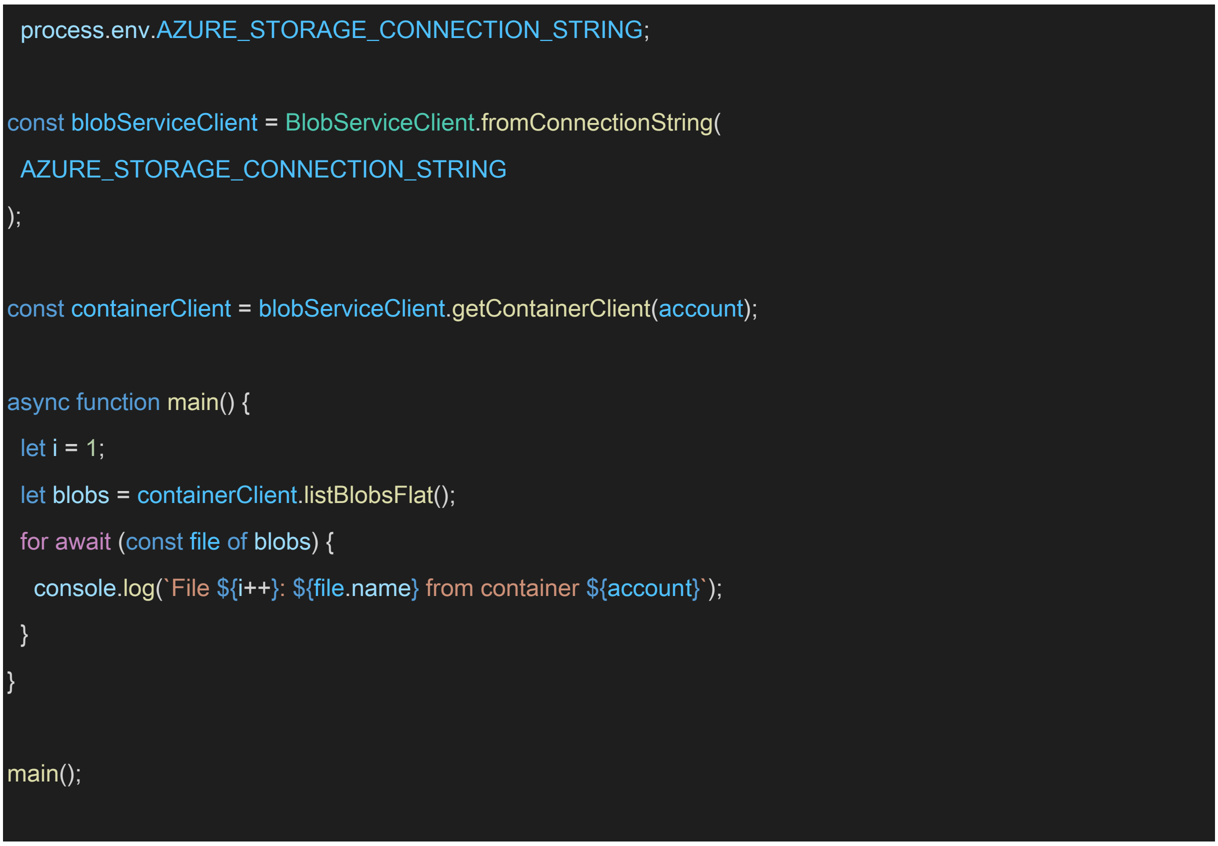
```
require("dotenv").config();

const { BlobServiceClient } = require("@azure/storage-blob");

const account = "mysamplecontainer-cse19650";

const AZURE_STORAGE_CONNECTION_STRING =
  process.env.AZURE_STORAGE_CONNECTION_STRING;

const blobServiceClient = BlobServiceClient.fromConnectionString(
  AZURE_STORAGE_CONNECTION_STRING
);

async function main() {
  let i = 1;
  let containers = blobServiceClient.listContainers();
  for await (const container of containers) {
    console.log(`Container ${i++} name : ${container.name}`);
  }
}

main();
```



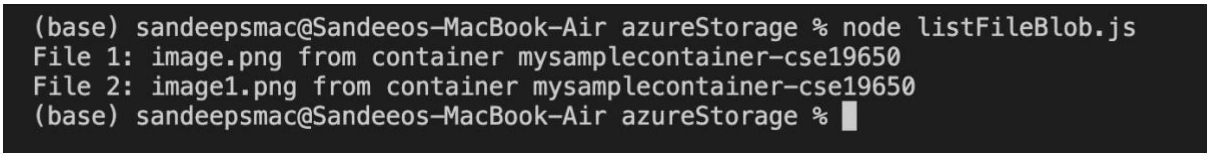
```
(base) sandeepsmac@Sandeeos-MacBook-Air azureStorage % node listContainers.js
Container 1 name : mysamplecontainer-cse19650
(base) sandeepsmac@Sandeeos-MacBook-Air azureStorage % █
```

8. List all files from a specific blob

```
require("dotenv").config();

const { BlobServiceClient } = require("@azure/storage-blob");

const account = "mysamplecontainer-cse19650";

const AZURE_STORAGE_CONNECTION_STRING =
```



```
  process.env.AZURE_STORAGE_CONNECTION_STRING;

const blobServiceClient = BlobServiceClient.fromConnectionString(
  AZURE_STORAGE_CONNECTION_STRING
);

const containerClient = blobServiceClient.getContainerClient(account);

async function main() {
  let i = 1;
  let blobs = containerClient.listBlobsFlat();
  for await (const file of blobs) {
    console.log(`File ${i++}: ${file.name} from container ${account}`);
  }
}

main();
```



```
(base) sandeepsmac@Sandeeos-MacBook-Air azureStorage % node listFileBlob.js
File 1: image.png from container mysamplecontainer-cse19650
File 2: image1.png from container mysamplecontainer-cse19650
(base) sandeepsmac@Sandeeos-MacBook-Air azureStorage % 
```